An extention of Grover's quantum search algorithm (Foundations of Computer Science)

AUTHOR(S):

Tanaka, Keisuke

CITATION:

Tanaka, Keisuke. An extention of Grover's quantum search algorithm (Foundations of Computer Science). 数理解析研究所講究録 2000, 1148: 47-51

ISSUE DATE:

2000-04

URL:

http://hdl.handle.net/2433/64016

数理解析研究所講究録
1148巻 2000年 47-51

An extention of Grover's quantum search algorithm

Keisuke Tanaka (田中 圭介)

NTT Information Sharing Platform Laboratories
Room 612A, 1-1 Hikarinooka Yokosuka, Kanagawa 239-0847, Japan
Phone: +81-468-59-4329 Fax: +81-468-59-3858
e-mail: keisuke@isl.ntt.co.jp

March 17, 2000

Abstract

Quantum computers and their algorithm are now widely studied. We consider the problem of making a superposition with an arbitrarily specified amplitudes. By modifying Grover's quantum search algorithm, we show that this can be done in time $O(\sqrt{N})$ with two simple operations, one is the Walsh–Hadamard transformation of $N \times N$-matrices where $N = 2^n$, the other is phase rotation of a state. If we see this problem in the classical computation model, it corresponds to the problem of making a sample with an arbitrarily specified distribution, where its time complexity is $\Omega(N)$. By making a measurement on the result of our algorithm, it is possible to obtain a sample according to an arbitrarily specified probability distribution in $O(\sqrt{N})$ time.

1 Introduction

In recent years, a significant progress has been made in the theory of quantum computation. For some particular problems including factoring integers and computing discrete logarithms, polynomial time algorithms on quantum computation are proposed [10]. These algorithms are exponentially faster than those known on classical computation. It is also known that, for finding an entry of target value in a database, there are quantum algorithms quadratic faster than classical ones [6].

We consider the problem of making a superposition with an arbitrarily specified amplitudes. This can suggests the elements or primitives for making quantum algorithms. If we see this problem in the classical computation model, it corresponds to the problem of making a sample with an arbitrarily specified distribution. Sampling is a basic technique of making probabilistic algorithms including primality testing [9] or approximating the permanent [2]. For some applications, uniform sampling is enough, but for some applications, sampling according to a specified (non-uniform) probability distribution is required. Classically, it is not possible to sample precisely according to an arbitrary probability distribution in fewer than $\Omega(N)$ time, where N is the number of elements. It is easily seen that we will need to examine at least half of the points, since, if we leave out half of the points, we could be missing a point with high probability. Recently, computational perspective on sampling is also considered [5].

Independently, sampling through quantum computing was proposed by Grover [8]. Our purpose is the same as his, and he gave his algorithm with generalized analysis in his paper. However, our method is totally different from his. We employ original Grover's transformations for database search, and propose a simpler algorithm (see also [7]). In fact, all of our operations are the Walsh–Hadamard transformations and phase rotations for $N \times N$-matrices where $N = 2^n$.

2 Problem

In this section, we formalize our problem. Let N be the size of input, and let a system have N states which are labelled $s_1, \ldots, s_N$. If $N = 2^n$ $(n = 1, 2, \ldots)$, then each s_i can be represented by n-bit strings. We specify the target arbitrary amplitude, a complex number, for each state s_i as input. The problem is to make a superposition described by input.

3 Algorithm

In this section, we suggest our algorithm. Before going into the main part of the algorithm, we need some initialization. The algorithm first splits each state into two, and makes a uniform distribution for these states.

1. To make the Walsh–Hadamard transformations easy, we normalize the size of input such that $N = 2^n$ $(n = 1, 2, \ldots)$. If $N \neq 2^n$, then we can add dummy entries until we have $N = 2^n$. The target amplitudes for dummy entries are 0. Notice that these N states are represented as n bit strings.

2. We denote the state of the system by (s_i, b), where s_i is a n-bit vector corresponded to the state and b is single additional bit used for the algorithm. Including an additional qubit, initialize the state so that all $n + 1$ qubits are 0.

3. We set the target amplitudes for (s_i, b), $b = 1, 2$, to be $\frac{1}{\sqrt{2}} \times |(\text{the initial target amplitude for } s_i)|$.

4. Initialize the system (including a single qubit) to the distribution:

$$(\frac{1}{\sqrt{2N}}, \frac{1}{\sqrt{2N}}, \ldots, \frac{1}{\sqrt{2N}}),$$

   i.e., there is the same amplitude in each of the $2N$ state. This distribution can be obtained in $O(\log N)$ time by standard techniques using the Walsh–Hadamard transformations (see [6]).

5. Let S_i is a set of states whose target amplitudes are the same value A. We denote this as $f(S_i) = A$. Let S be the set of all S_i's.

The main part of our algorithm is as follows. The algorithm now repeats Grover's original quantum search algorithms for making the amplitudes of target states bigger, with phase rotations for making establised states escaped

- Repeat the following until S is empty.
  1. Choose S_k such that $f(S_k) = \max_{S_i \in S} |f(S_i)|$.
  2. Repeat the following in the predetermined number of times described in the following section, until the amplitudes of the states in S_k reach their target values.
     (a) Rotate the phase of each state in S_k by π. Leave the other states as they are.
     (b) Let B be the average of all the amplitude of the states in S. Rotate the phase of each state $(s_i, 0)$ not in S by $\cos^{-1} \frac{B}{|(s_i,0)|}$. Rotate the phase of each state $(s_i, 1)$ not in S by $-\cos^{-1} \frac{B}{|(s_i,1)|}$.

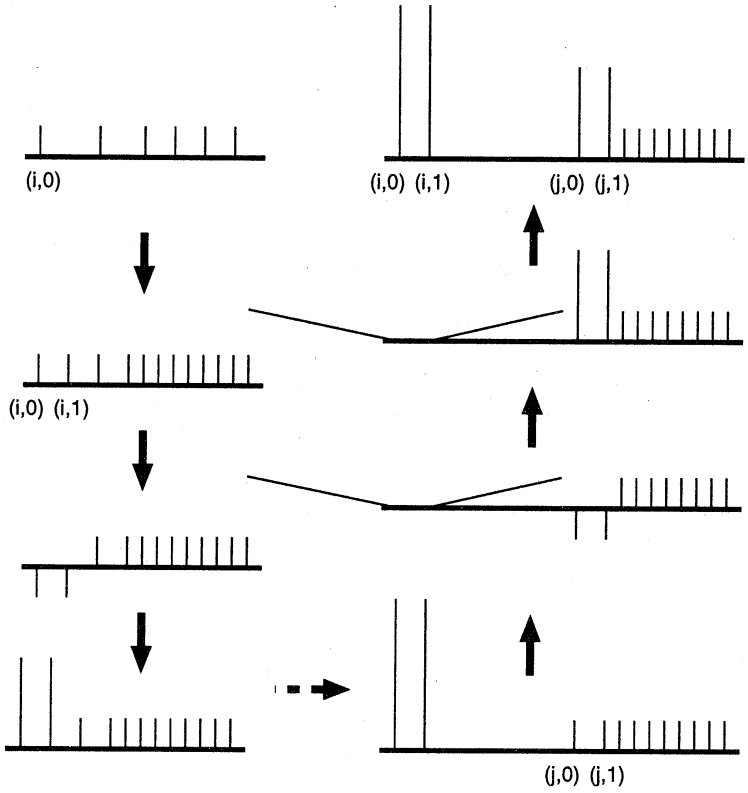

Figure 1: Algorithm

(c) Apply the diffusion transform D which is defined by the matrix D as follows:

$$D_{ij} = \frac{2}{N} \quad \text{if} \quad i \neq j,$$

$$D_{ii} = -1 + \frac{2}{N}.$$

(d) Rotate the phase of each state $(s_i, 0)$ not in S by $\cos^{-1} \frac{B}{|(s_i,0)|}$. Rotate the phase of each state $(s_i, 1)$ not in S by $-\cos^{-1} \frac{B}{|(s_i,1)|}$.

3. $S \longleftarrow S \setminus S_k$

After running the main part of the algorithm, we apply the Walsh–Hadamard transformation to an additional bit b in (s_i, b) to get only $(s_i, 0)$'s. Finally, rotate each phase of the state $(s_i, 0)$ as described by input to get the final result. If we measure this state, it is possible to obtain a sample according to an arbitrarily specified probability distribution. The picture of the algorithm is shown in Figure 1.

4 Analysis

First, we estimate the number of repetition of 2 of the main part of the algorithm. This can be determined by function appeared in [1]. In general case, with amortized analysis, the total number of repetition is $O(\sqrt{N})$.

Next, we see the diffusion transform D in (c) in the main part of the algorithm. As described in [6], D can be implemented as $D = WRW$, where R is a rotation matrix and W is the Walsh–Hadamard transform matrix. The rotation matrix R is defined as follows:

$$R_{ij} = 0 \quad \text{if} \quad i \neq j,$$

$$R_{00} = 1 \quad \text{and} \quad R_{ii} = -1 \quad \text{if} \quad i \neq 0.$$

The Walsh–Hadamard transform matrix W is defined as follows:

$$W_{ij} = 2^{-n/2}(-1)^{\bar{i}\cdot\bar{j}},$$

where $\bar{j}$ is the binary representation of i, and $\bar{i} \cdot \bar{j}$ denotes the bitwise dot product of the two binary strings $\bar{i}$ and $\bar{j}$.

It should be noticed that we only use the two types of operations, the Walsh–Hadamard transformations and phase rotations for $N \times N$-matrices where $N = 2^n$. The Walsh–Hadamard transformation for $N \times N$-matrices where $N = 2^n$ is very simple by using 2×2-matrix corresponding to the following operations:

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

$$|1\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

The simple way of the Walsh–Hadamard transformation of $N \times N$-matrices where $N \neq 2^n$ is not known (see also [4] for approximate Fourier transforms).

5 Concluding remarks

Independently, sampling through quantum computing was proposed by Grover [8]. Our purpose is the same as his, and he gave his algorithm with generalized analysis in his paper. However, our method is totally different from his.

We employ original Grover's transformations for database search, and propose a simpler algorithm (see also [7]). In fact, all of our operations are the Walsh–Hadamard transformations and phase rotations for $N \times N$-matrices where $N = 2^n$. This also has an advantage if we make experiments for the algorithms, since Grover's quantum search algorithm is one of the cases widely studied.

A disadvantage of our algorithm is that the phase rotations are quite complicated. However, this does not seem to be avoided if we employ original Grover's operations.

It is known that original Grover's search algorithm does not work well when the number of target entries are more than $N/4$. Chi and Kim [3] fix this situation by modifying the operations of the Walsh–Hadamard transforms as well as phase rotations. We can also fix this situation by applying our technique described above with the original Walsh–Hadamard transforms and rotations of π radian, combined with hash functions.

Acknowledgements

We would like to thank Tetsuro Nishino for introducing me to the field of quantum computation. We would also like to thank Tatsuaki Okamoto and Takashi Mihara for helpful discussion and valuable comments.

References

[1] BOYER, M., BRASSARD, G., HØYER, P., AND TAPP, A. Tight bounds on quantum searching. In *PhysComp* (1996).

[2] BRODER, A. How hard is to marry at random? (on the approximation of the permanent). In *Proceedings of ACM Symposium on Theory of Computing* (1986), pp. 50–58.

[3] CHI, D. P., AND KIM, J. Quantum database searching by a single query. quant-ph/9708005, 1997.

[4] COPPERSMITH, D. An approximate fourier transform useful in quantum factoring. IBM Research Report RC 19642, 1994.

[5] GOLDREICH, O. A sample of samplers—a computational perspective on sampling. ECCC TR97-020, 1997.

[6] GROVER, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of ACM Symposium on Theory of Computing* (1996).

[7] GROVER, L. K. A framework for fast quantum mechanical algorithms. In *Proceedings of ACM Symposium on Theory of Computing* (1998).

[8] GROVER, L. K. Rapid sampling through quantum computing. quant-ph/9912001, 1999.

[9] MILLER, G. Riemann's hypothesis and test for primality. *Journal of Computing and System Sciences 13*, 3 (1976), 300–317.

[10] SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26*, 5 (1997), 1484–1509.